

LIST

LONG ISLAND SINCLAIR TIMEX GROUP
 INCORPORATING * NYTSE OF NEW YORK CITY
 ISSUE: APRIL 1990



DISK DRIVES SUPPORTED:
 1 OLIBEE
 2 LARKEN
 3 AERCO

NEXT MEETING MAY 20

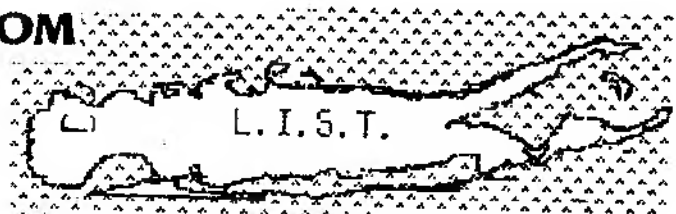


LIST MEMBERSHIP IS \$15.00. LIBRARY TAPES ARE AVAILABLE, WRITE THE
 ADDRESS PRINTED BELOW.



**A Cheap and Simple EPROM
 Programmer**

ZX81 Software



=====

L.I.S.T.
 5 PERI LANE
 VALLEY STREAM, NY
 11581

TO:

FIRST CLASS MAIL
 DATED MEETING NOTICE

UPPER RIGHT
 CORNER OF
 YOUR LABEL
 IS DATE OF
 LAST ISSUE.

LISTing

LIST OFFICERS

PRES. HARVEY RAIT
TRES. ROBERT MALLOY
REC. SEC. STEVE KAYE
EDITOR. FRED STERN
LIBR. TOM SKAPINSKI

PLEASE SEND INQUIRIES TO:

LIST
MR. HARVEY RAIT
5 PERI LANE
VALLEY STREAM, N.Y. 11581

PLEASE SEND SUBMISSIONS TO:

LISTING
MR. FREDERIC STERN
214 ROBERTS ST.
HOLBROOK, N.Y. 11741

NYTSE

NYTSE MEETS THE MONDAY AFTER
THE LIST MEETING AT:
MISS KIMS RESTAURANT
PARK AVENUE SOUTH
BETWEEN 21 ST. AND 22 ST.
MEETINGS START 7:30 PM.

COMING EVENTS:

MAY 20, 1990 LIST MEETING
MAY 21, 1990 NYTSE MEETING

MEETING MINUTES
APRIL 20, 1990

THIS IS THE FIRST MEETING HELD
AT THE WOODBURY NURSING HOME.

HARVEY CALLED THE MEETING TO
ORDER AT 2:15 PM.

HARVEY AND FRED READ LETTERS
RECEIVED SINCE THE LAST MEETING.
WE WELCOME 3 NEW MEMBERS;
WILLIE ROLDAN, BRONX N.Y.
MARK AND MARY YOST, CANTON CT.

HELP WANTED, HELP WANTED,

FRED STERN HAS BEEN ANSWERING
LETTERS TO LIST AND EDITING THE
NEWSLETTER. WE NEED A MEMBER TO
TAKE OVER ANSWERING THE
CORRESPONDENCE.
FRED WILL CONTINUE THE JOB UNTIL
THE SUMMER BREAK.
INTERESTED MEMBERS CAN CONTACT
EITHER FRED OR HARVEY.

NEWS, NEWS, NEWS, NEWS, NEWS,

WE RECEIVED A FLYER FROM BUDGET
ROBOTICS, TUCSON AZ. BUDGET IS
SELLING PARTS AND PERIPHERALS
FOR THE ZX-81 AND TS1000. OF
INTEREST ALSO IS AN AD FOR ZX-81
KITS THROUGH ZEERA SYSTEMS FOR
\$49.95.

MAR/APR 90 SINCUS REPORTS THAT
E. ARTHUR BROWN HAS AN AD ALSO
FOR THE ZX-81 KIT FOR \$39.95.

DEMONSTRATION

JIM DOUCAS AND BOB GILDER
DEMONSTRATED NETWORKING USING
2 OR MORE QL COMPUTERS. JIM
EXPLAINED THAT TWO PROGRAMS ARE
REQUIRED TO GET THE SYSTEM GOING
NETWORK ON TOOL KIT IS THE PRIME
PROGRAM, BUT E-SERVE IS ALSO
NEEDED TO CORRECT FAULTS IN THE
NET PROGRAM.

I HOPE JIM OR BOB WRITE AN
ARTICLE FOR LISTING TO BETTER
EXPLAIN THE SOFTWARE AND HOOKUP.

COMING EVENTS

OUR NEXT MEETING WILL AGAIN BE
HELD AT THE:
WOODBURY NURSING HOME
8533 JERICHO TURNPIKE
WOODBURY N.Y.

OUR JUNE 10, 1990 MEETING AND
SWAPMEET WILL BE AT HARVEY'S
HOUSE IN VALLEY STREAM.

SPECIAL THANK YOU

TO MS. KIM GIDIA AND THE STAFF
OF THE WOODBURY NURSING HOME
FOR THERE HOSPITALITY AND
EFFORTS TO MAKE OUR MEETING A
GREAT SUCCESS.

CLASSIFIEDS

THIS CLASSIFIED SECTION IS
AVAILABLE TO ALL LIST MEMBERS
FREE OF CHARGE.
THE ONLY RESTRICTION IS THAT
IT IS TO BE USED ONLY FOR THE
SEEKING, SELLING OR SWAPPING
OF SINCLAIR, TIMEX OR MICROACE
COMPUTER EQUIPMENT, PERIPHERALS
AND SOFTWARE.
LISTING, LIST, AND ITS OFFICERS
DO NOT ENDORSE, WARRANTY, OR
GUARANTEE ANY OF THE ITEMS
LISTED IN THIS CLASSIFIED
SECTION

IF YOU HAVE A COPY OF Q-SAVE,
FOR THE TS1000 PLEASE CONTACT
FRED STERN 516-737-0963.

A FINAL WORD

MY NAME IS FRED STERN AND I AM
THE EDITOR OF THIS EDITION OF
LISTING.

THANK YOUS GO TO TOM SKAPINSKI
FOR HIS ASSISTANCE.

ENCLOSED IN THIS ISSUE IS A
PARTIAL LIST OF OUR MEMBERSHIP.
NAMES AND ADDRESSES OF MANY
MEMBERS WERE NOT INCLUDED BY THE
MEMBERS REQUEST.

THE NOTICE OF THE KEN GORDON
SHOW MENTIONED IN THE LAST
NEWSLETTER WAS DELETED BEFORE
GOING TO PRINT FOR OBVIOUS
REASONS. SORRY FOR ANY
INCONVENIENCES.
SEE YOU ALL AT THE NEXT MEETING.

LONG ISLAND SINCLAIR TIMEX GROUP



Isidore Goldsmith AUG/90
537 Monaco L, Kings Point
Delray Beach, FL
33446

(305)495-4318

Harvey Rait JAN/91
5 Per1 Ln.
Valley Stream, NY
11581
(516) 791-6247

Bob Malloy JAN/91
412 Pacific St.
Massapequa Park, NY
11762

516-541-6731
Don Lambert JAN/91
3310 Clover Dr. S.W.
Cedar Rapids, IA
52404

Robert Gilder JAN/91
69 Jefferson Pl.
Massapequa, NY
11758

Jim Doucas JAN/90
1633 Crosby Rd.
Bronx, NY
10461

Hugo DiGiovanni JAN/90
6 Wallingford Dr.
Melville, NY
11747

Ike Walker JAN/91
253 Kenna Drive
So. Charleston W VA
25309-2646

304-346-2929

John Pazmino JAN/91
979 E 42 St.
Brooklyn, NY
11210

Stoney McMurray JAN/91
473 Westminster Rd
Brooklyn, NY
11218

(718)469-5948

Steve Kaye JAN/91
1281 E 86th St.
Brooklyn, NY
11236

Myles Cohen JAN/91
10 E 95 St.
New York, NY
10128

Ken Diederich NOV/90
312 N. Bailey
Jacksonville, AR
72076

Thomas Skapinski JAN/91
7 Atkinson Lane
Coram, NY
11727

(516)732-1825

Fred Stern JAN/91
214 Roberts St.
Holbrook, NY

11741
(516) 737-0963

Philip Florio JAN/91
28-35 Jordan St.
Bayside, NY
11358

Ed Lee JAN/91
17 w. 17th St.
New York, NY
10011

Joeseeph F. LaPunzina NOV/90
91 Bay 31 St
Brooklyn, NY
11214



Thomas W. Jennens MAR/91
33901 Lawton Ave.
Eastlake, OH
44095

John D. Solomon Jr. JAN/91
139 North Terrace Ave.
Mount Vernon, NY
10550

Charles J. Musumerci MAR/90
Coler Memorial Hosp. Ward A-42
Roosevelt Island, NY
10044

Don & Eleanor Lamen MAY/90
R.D. #3 Box 3404
Windsor, NY
13865

Mark Scheck JUN/90
86 Holmes St.
Statford, CT
06497

(203)378-4186

Charles R. Byler SEP/90
P.O. Box 5061
APO SF, CA
96519

Jay Siegel MAR/91
5213 9th Ave.
Brooklyn, NY
11220

(718)436-5367

Armand R. Martin OCT/90
431 Dubuc Street
Winnipeg, Manitoba
CANADA R2H 1G4

John J. Rossi III OCT/90
6 Beech Ct.
College Point, NY
11356

Jess Wyder DEC/90
17 Academy St.
Fishkill, NY
12524-1301

Nicholas I. Oshana, Jr. NOV/90
101 Treble Rd.
Bristol, CT
06010

Dr. Armand Drucker JAN/91
194-02A 67th Ave.
Fresh Meadows, NY
11356

Dr. Raymond Kaufman FEB/91
3755 Henry Hudson Pky. Apt 4F
Bronx, NY
10463

Richard K. Norek FEB/91
188 St. Felix Ave
Cheektowaga, NY
14227-1228

Robert M. Curnutt FEB/91
10400 Truxton Rd.
Adelphi, MD
20783

Paul W. Chomitz JAN/91
95 Delancey St.
New York, NY
10002

LIST.
LIST.

TRY THIS

A Cheap and Simple EPROM Programmer

Dan Schaaf



"What more could I want?" That's what you usually say before you buy something. But you learn after you have had it for a while that there is a lot more to be wanted. Past

Dan Schaaf, 306 N. Carroll, Michigan City, IN 46360.

issues of *SYNC* have had several articles to remedy heart-felt deficiencies in the Sinclair Basics: SCROLLs, PAUSEs, READs, DATAs, string handling routines, etc.

This article will show you how to build an add-on device which will allow you to have any of these routines, plus any you can design yourself, permanently and instantly available to you in one 4K EPROM without the need to

LOAD those routines from tape into the machine (a waste of time and of RAM). Those interested in peripherals for the Sinclairs (floppy disk, electronic typewriters, etc.) can design monitors and controllers for those devices. Eventually you can program independent microprocessors for special purposes. An EPROM programmer can become a real liberator of the power inherent in the Z80.

4K EPROMs now cost less than \$10.00. 4K was chosen because of its cost advantage and simplicity of programming. 4K is a good healthy chunk of memory as witnessed by the

Figure 1. Schematic Diagram.

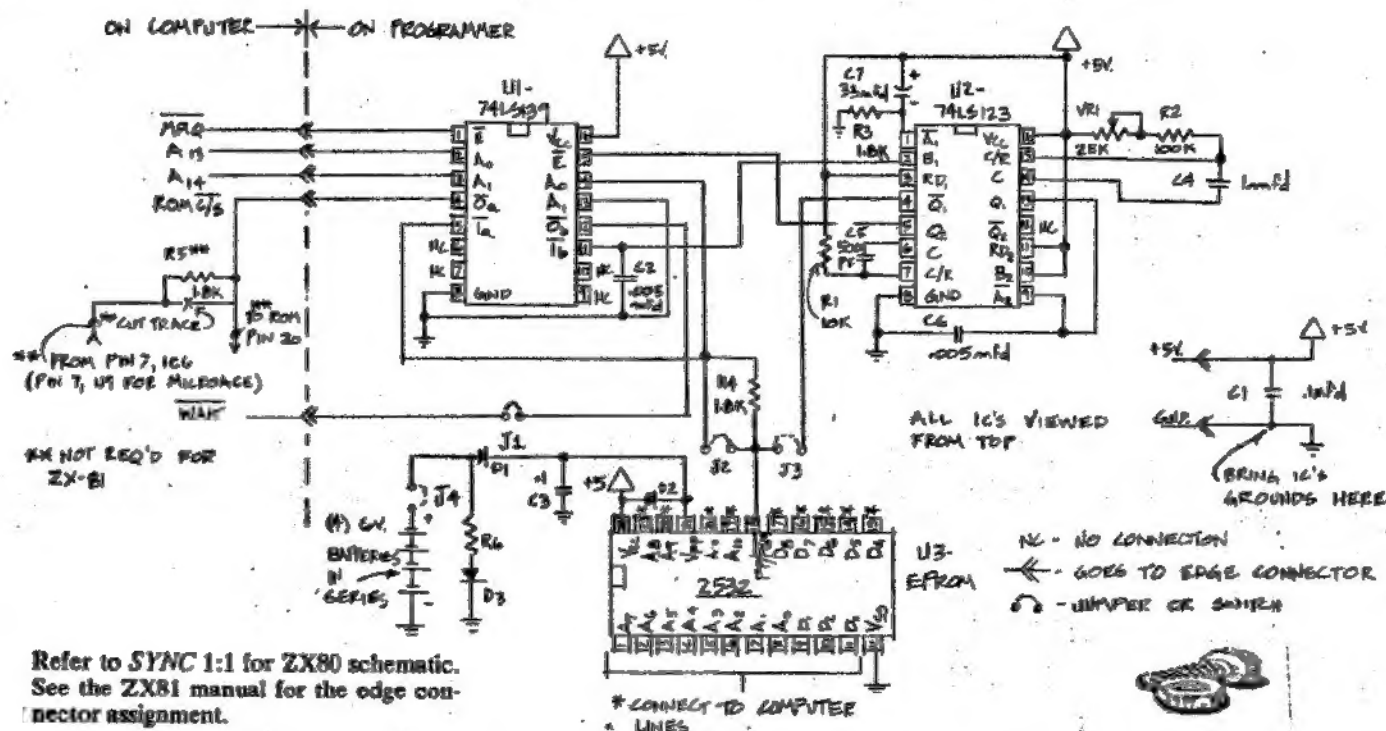


Figure 2. Timing Diagram.

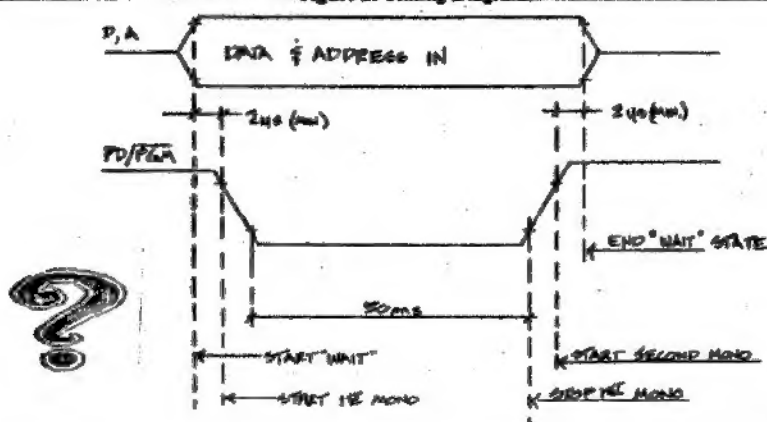
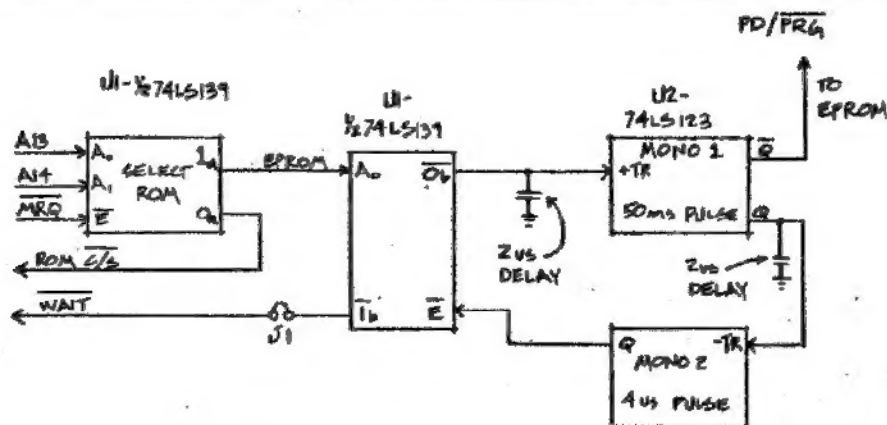


Figure 3. Block Diagram.



The Block Diagram shows how the basic EPROM timing needs are met by the circuit. As soon as a Memory Request is made into the EPROM, a Wait state is initiated. This holds data and addresses on the bus line until cleared by the timing loop. The beginning of

the 50 ms pulse to the PD/PGM pin is delayed 2 μ s (capacitor C2). 2 μ s (capacitor C6) after the 50 ms pulse ends, a 4 μ s pulse from the second mono clears the Wait state and allows the computer's bus lines to operate as normal.

Sinclair 4K Basic. Although not necessary since EPROMs come "erased" from the factory, it is useful to have or have access to an ultraviolet lamp for erasing the EPROM. They can be bought new for \$70.00, but some can be found at flea markets and are sometimes known as rock hound lights used to fluoresce organic and other materials in rocks. At \$70.00, it should be noted, however, seven fresh EPROMs could be bought.

Once built, programming proceeds by POKEing data into the proper addresses. USR is then used if the data was a machine language program, or PEEK if the data was a table, and also PEEK to check that programming worked properly.

What Is an EPROM?

EPROM stands for Erasable Programmable Read Only Memory. An EPROM has some of the same characteristics as the ROM in your machine that contains the Basic language:

1) Data may be read from the EPROM once an address has been presented to the address bus.

2) Data is retained when the power is turned off, almost forever.

There are differences: EPROMs need to be programmed a byte at a time; ROMs are made in large batches each with the same program on them. EPROMs can be erased (for reprogramming) by the use of ultraviolet radiation; ROMs can never be erased. In a sense, an EPROM is midway between a ROM and a RAM. Typically an EPROM has the same gross dimensions as a ROM and the same pin-out as a ROM (since the computer uses both the same way) but in addition it has a little quartz window on top that allows the ultraviolet light into the EPROM for erasing. Erasing results in all data being lost at all addresses. There is no selective erasing.

How Does an EPROM Work?

The basic element of the EPROM is a capacitor which holds a charge. The estimated life time (time constant) of the charge is 7 to 10 years. Each bit is represented within the EPROM by one capacitor. A sensing amplifier converts this static charge to a TTL

level voltage that appears on the data lines as either a "1" or "0". Ultraviolet radiation has sufficient energy to dislodge electrons on the capacitor surface, and as these photo-electrons are generated the capacitor will discharge and data will be erased. Most EPROMs read a discharged capacitor as a "1" and a charged capacitor as a "0". 25 volts are used to charge the 32,768 tiny capacitors on the chip. A typical ultraviolet lamp takes about half an hour to discharge the EPROM. Manufacturers advise that a tape cover be put over the window to prevent accidental erasure.

How the EPROM Programmer Works

Looking at the data sheets for the EPROM selected, one sees various requirements:

- 1) Data and address information must be at TTL compatible voltages.
- 2) For programming, 25 volts must be applied to pin 21 (V_{pp}).
- 3) Data and addresses must be held constant for 50 milli-seconds.
- 4) A certain timing sequence must be carried out with regards to the activation of the programming pin, pin 18 (See Timing Diagram.)

1) and 2) are easily met. The Z80 has TTL compatible voltage levels and four 6 volt batteries will provide 25 volts. 3) and 4) are harder to meet. A POKE command only holds address and data information on the busses for about 550 nanoseconds, almost 100,000 times too short for proper EPROM programming.

The solution is to use the WAIT pin on the Z80. When the WAIT pin is active (low voltage) all bus information is held constant until the pin voltage goes high again. The circuit shown has a one-shot whose time period is set to 50 microseconds. The other timing requirements are taken care of by the circuit as well, as explained next.

When the EPROM is addressed (a proper combination of A13, A14, and MRQ, memory request) by the Z80, U1 switches the internal ROM off (via \bar{O}_B -pin 4) and turns on the EPROM (via \bar{I}_B -pin 5). If J1 is in, a WAIT state is started when \bar{O}_B pin 12 of U1 goes low. At the same time, pin 11 \bar{I}_B of U1 goes high initiating the 50 millisecond one-shot of U2 (VR1 + R2, C4). Q1 of U2 goes low for 50 milliseconds as does PGM, pin 18 of U3, which then programs the EPROM. 2 milliseconds after the end of the 50 millisecond programming period, the second one-shot of U2 (R1, C5) disables the Enable pin, pin 15, of U1 thus clearing the WAIT state and allowing the computer to process the next instruction.

The design was such as to use a minimum of parts. Certain problems therefore exist which can be avoided by being aware of them. The main concern is that a PEEK command, or USR command into the EPROM when the EPROM is in the "program" mode will write useless data into the EPROM. Therefore, do not PEEK or USR into the EPROM without disconnecting the 25 volts (J4 out).

If you are familiar with dynamic RAM specifications (the type of RAM used in the 16K-64K expansion units), you will know that they specify a maximum refresh period usually of 2 milliseconds. During a WAIT cycle there is no refreshing. So if the data sheets are to be believed, dynamic RAMs should not work with this EPROM. However, I have found that dynamic RAMs can tolerate as much as one or two seconds without refresh and without loss of data. I have used my EPROM programmer with 16K and the newer 64K dynamic RAMs and have had no troubles. If you run into trouble, disconnect the dynamic RAMs and use the internal static RAMs, which need no refresh.

Making It

The construction method described here is the one I used and may not be best for you.

What I usually do is lay out the ICs (or IC sockets) on a board, making sure that there is enough room for miscellaneous components and that the wiring which will interface with the computer is conveniently close to the edge of the board. I seek to minimize runs, especially power runs (5 and 25 volt). I then put in the other components and use the leads as much as possible to wire the circuit. Every step of the way I check and double check that my wiring corresponds to the schematic. ICs get confusing since you must look at them from the top and the bottom. (See schematic.)

Many years ago I got a lot of color-coded telephone wire (what more could I want?) from an abandoned factory in my home town.

I use that to wire the rest of the circuit. I recheck the circuit against the schematic and then solder all the points. After that I use a Volt-Ohm Meter to test the continuity of each run as well as that two runs have not accidentally shorted. Good, clean, logical construction habits save hours of needless trouble shooting later. I try to document what and how I have wired it after I am done.

So now you have a board wired as in the schematic. Next it must be interfaced with the computer. Here again I used a method which may not be right for you. Any method is acceptable, including hardwiring it with color coded telephone wire. However, this could limit future expansion. AP Products (as shown on page 42 of the 1982 Jameco Cata-

log) has jumper headers and cable assemblies which can be used. I used two straight double row, 40 post jumper headers. One set I soldered onto the EPROM board. The other set I soldered onto the computer's edge connector lower forty.

I then interconnected the two boards with a 6" AP flat ribbon cable. Long runs of data lines can cause problems and should be avoided. Of the four remaining edge connector positions, only two are needed: D7 and 5+ volts. These I hard wired with a connector in between for disconnecting.

The EPROM is a MOS device and is susceptible to damage due to static discharge. I have not been overly cautious, yet I have never had trouble with static charge destroying a

Figure 4. Parts List.

Part	Radio Shack Number. (If more than one of item in package, then number is listed only the number of times needed.)
ICs	
U1	74LS139
U2	74LS123
U3	TMS 2532, 450 ns.
Resistors	
R1	10K
R2	100K
R3	1.8K
R4	1.8K
R5	1.8K
R6	3.3, 1/2 watt (optional)
Capacitors	
C1	.1 mfd
C2	.005 mfd
C3	.1 mfd
C4	1 mfd
C5	470 pfd
C6	.005 mfd
C7	33 mfd, 16V.
Diodes	
D1	Diode
D2	Diode
D3	LED (optional)
Variable Resistor	
VR1	25K
Miscellaneous	
J1, J4	Jumpers or one way switches
J2-3	3-post jumper or 2-way switch
PC Board	276-153
IC socket for U3 - 24 pin*	276-1989
Batteries (4) - 6 volt lantern	Eveready 509 or equal
Connector Cable - (See article "Making It")	

* The other IC's can be soldered into place without too much fear of damage or having to replace them. But the EPROM, and any MOS device, really should be in a socket.

** Available through many mail order houses, such as Jameco, Active Electronics, JDR Microdevices, etc.. Addresses and prices can be found in the back of most hobby electronics magazines.

```

Program 1.
10 FOR L=0 TO 255
20 POKE 16383,L
30 NEXT L

Program 2.
10 FOR L=0 TO 4095
20 IF L=0 THEN L=16383:GOTO 30
30 PRINT L:POKE L,L
40 NEXT L

Program 3.
10 IF INKEY$="P" THEN GOTO 5
20 POKE 16383,0
30 SAVE "P"
40 GOTO 20

Program 4.
SAVE AS PROGRAM 2

Program 5.
10 IF NOT INKEY$="P" THEN GOTO 5
20 POKE M,N1
30 POKE M+1,N2
40 POKE M+2,N3
50 POKE M+3,N4
60 POKE M+4,N5
70 POKE M+5,N6
80 POKE M+6,N7
90 POKE M+7,N8
100 POKE M+8,N9
110 POKE M+9,N10
120 POKE M+10,N11
130 POKE M+11,N12
140 POKE M+12,N13
150 POKE M+13,N14
160 POKE M+14,N15
170 POKE M+15,N16
180 POKE M+16,N17
190 POKE M+17,N18
200 POKE M+18,N19
210 POKE M+19,N20
220 POKE M+20,N21
230 POKE M+21,N22
240 POKE M+22,N23
250 POKE M+23,N24
260 POKE M+24,N25
270 POKE M+25,N26
280 POKE M+26,N27
290 POKE M+27,N28
300 POKE M+28,N29
310 POKE M+29,N30
320 POKE M+30,N31
330 POKE M+31,N32
340 POKE M+32,N33
350 POKE M+33,N34
360 POKE M+34,N35
370 POKE M+35,N36
380 POKE M+36,N37
390 POKE M+37,N38
400 POKE M+38,N39
410 POKE M+39,N40
420 POKE M+40,N41
430 POKE M+41,N42
440 POKE M+42,N43
450 POKE M+43,N44
460 POKE M+44,N45
470 POKE M+45,N46
480 POKE M+46,N47
490 POKE M+47,N48
500 POKE M+48,N49
510 POKE M+49,N50
520 POKE M+50,N51
530 POKE M+51,N52
540 POKE M+52,N53
550 POKE M+53,N54
560 POKE M+54,N55
570 POKE M+55,N56
580 POKE M+56,N57
590 POKE M+57,N58
600 POKE M+58,N59
610 POKE M+59,N60
620 POKE M+60,N61
630 POKE M+61,N62
640 POKE M+62,N63
650 POKE M+63,N64
660 POKE M+64,N65
670 POKE M+65,N66
680 POKE M+66,N67
690 POKE M+67,N68
700 POKE M+68,N69
710 POKE M+69,N70
720 POKE M+70,N71
730 POKE M+71,N72
740 POKE M+72,N73
750 POKE M+73,N74
760 POKE M+74,N75
770 POKE M+75,N76
780 POKE M+76,N77
790 POKE M+77,N78
800 POKE M+78,N79
810 POKE M+79,N80
820 POKE M+80,N81
830 POKE M+81,N82
840 POKE M+82,N83
850 POKE M+83,N84
860 POKE M+84,N85
870 POKE M+85,N86
880 POKE M+86,N87
890 POKE M+87,N88
900 POKE M+88,N89
910 POKE M+89,N90
920 POKE M+90,N91
930 POKE M+91,N92
940 POKE M+92,N93
950 POKE M+93,N94
960 POKE M+94,N95
970 POKE M+95,N96
980 POKE M+96,N97
990 POKE M+97,N98
1000 POKE M+98,N99
1010 POKE M+99,N100
1020 POKE M+100,N101
1030 POKE M+101,N102
1040 POKE M+102,N103
1050 POKE M+103,N104
1060 POKE M+104,N105
1070 POKE M+105,N106
1080 POKE M+106,N107
1090 POKE M+107,N108
1100 POKE M+108,N109
1110 POKE M+109,N110
1120 POKE M+110,N111
1130 POKE M+111,N112
1140 POKE M+112,N113
1150 POKE M+113,N114
1160 POKE M+114,N115
1170 POKE M+115,N116
1180 POKE M+116,N117
1190 POKE M+117,N118
1200 POKE M+118,N119
1210 POKE M+119,N120
1220 POKE M+120,N121
1230 POKE M+121,N122
1240 POKE M+122,N123
1250 POKE M+123,N124
1260 POKE M+124,N125
1270 POKE M+125,N126
1280 POKE M+126,N127
1290 POKE M+127,N128
1300 POKE M+128,N129
1310 POKE M+129,N130
1320 POKE M+130,N131
1330 POKE M+131,N132
1340 POKE M+132,N133
1350 POKE M+133,N134
1360 POKE M+134,N135
1370 POKE M+135,N136
1380 POKE M+136,N137
1390 POKE M+137,N138
1400 POKE M+138,N139
1410 POKE M+139,N140
1420 POKE M+140,N141
1430 POKE M+141,N142
1440 POKE M+142,N143
1450 POKE M+143,N144
1460 POKE M+144,N145
1470 POKE M+145,N146
1480 POKE M+146,N147
1490 POKE M+147,N148
1500 POKE M+148,N149
1510 POKE M+149,N150
1520 POKE M+150,N151
1530 POKE M+151,N152
1540 POKE M+152,N153
1550 POKE M+153,N154
1560 POKE M+154,N155
1570 POKE M+155,N156
1580 POKE M+156,N157
1590 POKE M+157,N158
1600 POKE M+158,N159
1610 POKE M+159,N160
1620 POKE M+160,N161
1630 POKE M+161,N162
1640 POKE M+162,N163
1650 POKE M+163,N164
1660 POKE M+164,N165
1670 POKE M+165,N166
1680 POKE M+166,N167
1690 POKE M+167,N168
1700 POKE M+168,N169
1710 POKE M+169,N170
1720 POKE M+170,N171
1730 POKE M+171,N172
1740 POKE M+172,N173
1750 POKE M+173,N174
1760 POKE M+174,N175
1770 POKE M+175,N176
1780 POKE M+176,N177
1790 POKE M+177,N178
1800 POKE M+178,N179
1810 POKE M+179,N180
1820 POKE M+180,N181
1830 POKE M+181,N182
1840 POKE M+182,N183
1850 POKE M+183,N184
1860 POKE M+184,N185
1870 POKE M+185,N186
1880 POKE M+186,N187
1890 POKE M+187,N188
1900 POKE M+188,N189
1910 POKE M+189,N190
1920 POKE M+190,N191
1930 POKE M+191,N192
1940 POKE M+192,N193
1950 POKE M+193,N194
1960 POKE M+194,N195
1970 POKE M+195,N196
1980 POKE M+196,N197
1990 POKE M+197,N198
2000 POKE M+198,N199
2010 POKE M+199,N200
2020 POKE M+200,N201
2030 POKE M+201,N202
2040 POKE M+202,N203
2050 POKE M+203,N204
2060 POKE M+204,N205
2070 POKE M+205,N206
2080 POKE M+206,N207
2090 POKE M+207,N208
2100 POKE M+208,N209
2110 POKE M+209,N210
2120 POKE M+210,N211
2130 POKE M+211,N212
2140 POKE M+212,N213
2150 POKE M+213,N214
2160 POKE M+214,N215
2170 POKE M+215,N216
2180 POKE M+216,N217
2190 POKE M+217,N218
2200 POKE M+218,N219
2210 POKE M+219,N220
2220 POKE M+220,N221
2230 POKE M+221,N222
2240 POKE M+222,N223
2250 POKE M+223,N224
2260 POKE M+224,N225
2270 POKE M+225,N226
2280 POKE M+226,N227
2290 POKE M+227,N228
2300 POKE M+228,N229
2310 POKE M+229,N230
2320 POKE M+230,N231
2330 POKE M+231,N232
2340 POKE M+232,N233
2350 POKE M+233,N234
2360 POKE M+234,N235
2370 POKE M+235,N236
2380 POKE M+236,N237
2390 POKE M+237,N238
2400 POKE M+238,N239
2410 POKE M+239,N240
2420 POKE M+240,N241
2430 POKE M+241,N242
2440 POKE M+242,N243
2450 POKE M+243,N244
2460 POKE M+244,N245
2470 POKE M+245,N246
2480 POKE M+246,N247
2490 POKE M+247,N248
2500 POKE M+248,N249
2510 POKE M+249,N250
2520 POKE M+250,N251
2530 POKE M+251,N252
2540 POKE M+252,N253
2550 POKE M+253,N254
2560 POKE M+254,N255
2570 POKE M+255,N256
2580 POKE M+256,N257
2590 POKE M+257,N258
2600 POKE M+258,N259
2610 POKE M+259,N260
2620 POKE M+260,N261
2630 POKE M+261,N262
2640 POKE M+262,N263
2650 POKE M+263,N264
2660 POKE M+264,N265
2670 POKE M+265,N266
2680 POKE M+266,N267
2690 POKE M+267,N268
2700 POKE M+268,N269
2710 POKE M+269,N270
2720 POKE M+270,N271
2730 POKE M+271,N272
2740 POKE M+272,N273
2750 POKE M+273,N274
2760 POKE M+274,N275
2770 POKE M+275,N276
2780 POKE M+276,N277
2790 POKE M+277,N278
2800 POKE M+278,N279
2810 POKE M+279,N280
2820 POKE M+280,N281
2830 POKE M+281,N282
2840 POKE M+282,N283
2850 POKE M+283,N284
2860 POKE M+284,N285
2870 POKE M+285,N286
2880 POKE M+286,N287
2890 POKE M+287,N288
2900 POKE M+288,N289
2910 POKE M+289,N290
2920 POKE M+290,N291
2930 POKE M+291,N292
2940 POKE M+292,N293
2950 POKE M+293,N294
2960 POKE M+294,N295
2970 POKE M+295,N296
2980 POKE M+296,N297
2990 POKE M+297,N298
3000 POKE M+298,N299
3010 POKE M+299,N300
3020 POKE M+300,N301
3030 POKE M+301,N302
3040 POKE M+302,N303
3050 POKE M+303,N304
3060 POKE M+304,N305
3070 POKE M+305,N306
3080 POKE M+306,N307
3090 POKE M+307,N308
3100 POKE M+308,N309
3110 POKE M+309,N310
3120 POKE M+310,N311
3130 POKE M+311,N312
3140 POKE M+312,N313
3150 POKE M+313,N314
3160 POKE M+314,N315
3170 POKE M+315,N316
3180 POKE M+316,N317
3190 POKE M+317,N318
3200 POKE M+318,N319
3210 POKE M+319,N320
3220 POKE M+320,N321
3230 POKE M+321,N322
3240 POKE M+322,N323
3250 POKE M+323,N324
3260 POKE M+324,N325
3270 POKE M+325,N326
3280 POKE M+326,N327
3290 POKE M+327,N328
3300 POKE M+328,N329
3310 POKE M+329,N330
3320 POKE M+330,N331
3330 POKE M+331,N332
3340 POKE M+332,N333
3350 POKE M+333,N334
3360 POKE M+334,N335
3370 POKE M+335,N336
3380 POKE M+336,N337
3390 POKE M+337,N338
3400 POKE M+338,N339
3410 POKE M+339,N340
3420 POKE M+340,N341
3430 POKE M+341,N342
3440 POKE M+342,N343
3450 POKE M+343,N344
3460 POKE M+344,N345
3470 POKE M+345,N346
3480 POKE M+346,N347
3490 POKE M+347,N348
3500 POKE M+348,N349
3510 POKE M+349,N350
3520 POKE M+350,N351
3530 POKE M+351,N352
3540 POKE M+352,N353
3550 POKE M+353,N354
3560 POKE M+354,N355
3570 POKE M+355,N356
3580 POKE M+356,N357
3590 POKE M+357,N358
3600 POKE M+358,N359
3610 POKE M+359,N360
3620 POKE M+360,N361
3630 POKE M+361,N362
3640 POKE M+362,N363
3650 POKE M+363,N364
3660 POKE M+364,N365
3670 POKE M+365,N366
3680 POKE M+366,N367
3690 POKE M+367,N368
3700 POKE M+368,N369
3710 POKE M+369,N370
3720 POKE M+370,N371
3730 POKE M+371,N372
3740 POKE M+372,N373
3750 POKE M+373,N374
3760 POKE M+374,N375
3770 POKE M+375,N376
3780 POKE M+376,N377
3790 POKE M+377,N378
3800 POKE M+378,N379
3810 POKE M+379,N380
3820 POKE M+380,N381
3830 POKE M+381,N382
3840 POKE M+382,N383
3850 POKE M+383,N384
3860 POKE M+384,N385
3870 POKE M+385,N386
3880 POKE M+386,N387
3890 POKE M+387,N388
3900 POKE M+388,N389
3910 POKE M+389,N390
3920 POKE M+390,N391
3930 POKE M+391,N392
3940 POKE M+392,N393
3950 POKE M+393,N394
3960 POKE M+394,N395
3970 POKE M+395,N396
3980 POKE M+396,N397
3990 POKE M+397,N398
4000 POKE M+398,N399
4010 POKE M+399,N400
4020 POKE M+400,N401
4030 POKE M+401,N402
4040 POKE M+402,N403
4050 POKE M+403,N404
4060 POKE M+404,N405
4070 POKE M+405,N406
4080 POKE M+406,N407
4090 POKE M+407,N408
4100 POKE M+408,N409
4110 POKE M+409,N410
4120 POKE M+410,N411
4130 POKE M+411,N412
4140 POKE M+412,N413
4150 POKE M+413,N414
4160 POKE M+414,N415
4170 POKE M+415,N416
4180 POKE M+416,N417
4190 POKE M+417,N418
4200 POKE M+418,N419
4210 POKE M+419,N420
4220 POKE M+420,N421
4230 POKE M+421,N422
4240 POKE M+422,N423
4250 POKE M+423,N424
4260 POKE M+424,N425
4270 POKE M+425,N426
4280 POKE M+426,N427
4290 POKE M+427,N428
4300 POKE M+428,N429
4310 POKE M+429,N430
4320 POKE M+430,N431
4330 POKE M+431,N432
4340 POKE M+432,N433
4350 POKE M+433,N434
4360 POKE M+434,N435
4370 POKE M+435,N436
4380 POKE M+436,N437
4390 POKE M+437,N438
4400 POKE M+438,N439
4410 POKE M+439,N440
4420 POKE M+440,N441
4430 POKE M+441,N442
4440 POKE M+442,N443
4450 POKE M+443,N444
4460 POKE M+444,N445
4470 POKE M+445,N446
4480 POKE M+446,N447
4490 POKE M+447,N448
4500 POKE M+448,N449
4510 POKE M+449,N450
4520 POKE M+450,N451
4530 POKE M+451,N452
4540 POKE M+452,N453
4550 POKE M+453,N454
4560 POKE M+454,N455
4570 POKE M+455,N456
4580 POKE M+456,N457
4590 POKE M+457,N458
4600 POKE M+458,N459
4610 POKE M+459,N460
4620 POKE M+460,N461
4630 POKE M+461,N462
4640 POKE M+462,N463
4650 POKE M+463,N464
4660 POKE M+464,N465
4670 POKE M+465,N466
4680 POKE M+466,N467
4690 POKE M+467,N468
4700 POKE M+468,N469
4710 POKE M+469,N470
4720 POKE M+470,N471
4730 POKE M+471,N472
4740 POKE M+472,N473
4750 POKE M+473,N474
4760 POKE M+474,N475
4770 POKE M+475,N476
4780 POKE M+476,N477
4790 POKE M+477,N478
4800 POKE M+478,N479
4810 POKE M+479,N480
4820 POKE M+480,N481
4830 POKE M+481,N482
4840 POKE M+482,N483
4850 POKE M+483,N484
4860 POKE M+484,N485
4870 POKE M+485,N486
4880 POKE M+486,N487
4890 POKE M+487,N488
4900 POKE M+488,N489
4910 POKE M+489,N490
4920 POKE M+490,N491
4930 POKE M+491,N492
4940 POKE M+492,N493
4950 POKE M+493,N494
4960 POKE M+494,N495
4970 POKE M+495,N496
4980 POKE M+496,N497
4990 POKE M+497,N498
5000 POKE M+498,N499
5010 POKE M+499,N500
5020 POKE M+500,N501
5030 POKE M+501,N502
5040 POKE M+502,N503
5050 POKE M+503,N504
5060 POKE M+504,N505
5070 POKE M+505,N506
5080 POKE M+506,N507
5090 POKE M+507,N508
5100 POKE M+508,N509
5110 POKE M+509,N510
5120 POKE M+510,N511
5130 POKE M+511,N512
5140 POKE M+512,N513
5150 POKE M+513,N514
5160 POKE M+514,N515
5170 POKE M+515,N516
5180 POKE M+516,N517
5190 POKE M+517,N518
5200 POKE M+518,N519
5210 POKE M+519,N520
5220 POKE M+520,N521
5230 POKE M+521,N522
5240 POKE M+522,N523
5250 POKE M+523,N524
5260 POKE M+524,N525
5270 POKE M+525,N526
5280 POKE M+526,N527
5290 POKE M+527,N528
5300 POKE M+528,N529
5310 POKE M+529,N530
5320 POKE M+530,N531
5330 POKE M+531,N532
5340 POKE M+532,N533
5350 POKE M+533,N534
5360 POKE M+534,N535
5370 POKE M+535,N536
5380 POKE M+536,N537
5390 POKE M+537,N538
5400 POKE M+538,N539
5410 POKE M+539,N540
5420 POKE M+540,N541
5430 POKE M+541,N542
5440 POKE M+542,N543
5450 POKE M+543,N544
5460 POKE M+544,N545
5470 POKE M+545,N546
5480 POKE M+546,N547
5490 POKE M+547,N548
5500 POKE M+548,N549
5510 POKE M+549,N550
5520 POKE M+550,N551
5530 POKE M+551,N552
5540 POKE M+552,N553
5550 POKE M+553,N554
5560 POKE M+554,N555
5570 POKE M+555,N556
5580 POKE M+556,N557
5590 POKE M+557,N558
5600 POKE M+558,N559
5610 POKE M+559,N560
5620 POKE M+560,N561
5630 POKE M+561,N562
5640 POKE M+562,N563
5650 POKE M+563,N564
5660 POKE M+564,N565
5670 POKE M+565,N566
5680 POKE M+566,N567
5690 POKE M+567,N568
5700 POKE M+568,N569
5710 POKE M+569,N570
5720 POKE M+570,N571
5730 POKE M+571,N572
5740 POKE M+572,N573
5750 POKE M+573,N574
5760 POKE M+574,N575
5770 POKE M+575,N576
5780 POKE M+576,N577
5790 POKE M+577,N578
5800 POKE M+578,N579
5810 POKE M+579,N580
5820 POKE M+580,N581
5830 POKE M+581,N582
5840 POKE M+582,N583
5850 POKE M+583,N584
5860 POKE M+584,N585
5870 POKE M+585,N586
5880 POKE M+586,N587
5890 POKE M+587,N588
5900 POKE M+588,N589
5910 POKE M+589,N590
5920 POKE M+590,N591
5930 POKE M+591,N592
5940 POKE M+592,N593
5950 POKE M+593,N594
5960 POKE M+594,N595
5970 POKE M+595,N596
5980 POKE M+596,N597
5990 POKE M+597,N598
6000 POKE M+598,N599
6010 POKE M+599,N600
6020 POKE M+600,N601
6030 POKE M+601,N602
6040 POKE M+602,N603
6050 POKE M+603,N604
6060 POKE M+604,N605
6070 POKE M+605,N606
6080 POKE M+606,N607
6090 POKE M+607,N608
6100 POKE M+608,N609
6110 POKE M+609,N610
6120 POKE M+610,N611
6130 POKE M+611,N612
6140 POKE M+612,N613
6150 POKE M+613,N614
6160 POKE M+614,N615
6170 POKE M+615,N616
6180 POKE M+616,N617
6190 POKE M+617,N618
6200 POKE M+618,N619
6210 POKE M+619,N620
6220 POKE M+620,N621
6230 POKE M+621,N622
6240 POKE M+622,N623
6250 POKE M+623,N624
6260 POKE M+624,N625
6270 POKE M+625,N626
6280 POKE M+626,N627
6290 POKE M+627,N628
6300 POKE M+628,N629
6310 POKE M+629,N630
6320 POKE M+630,N631
6330 POKE M+631,N632
6340 POKE M+632,N633
6350 POKE M+633,N634
6360 POKE M+634,N635
6370 POKE M+635,N636
6380 POKE M+636,N637
6390 POKE M+637,N638
6400 POKE M+638,N639
6410 POKE M+639,N640
6420 POKE M+640,N641
6430 POKE M+641,N642
6440 POKE M+642,N643
6450 POKE M+643,N644
6460 POKE M+644,N645
6470 POKE M+645,N646
6480 POKE M+646,N647
6490 POKE M+647,N648
6500 POKE M+648,N649
6510 POKE M+649,N650
6520 POKE M+650,N651
65
```

MOS device. One should, however, make sure before handling the EPROM that you are discharged to the ground of the computer board. Do not place EPROM into socket until both the timing and the 25v circuit have been checked out.

In making the EPROM programmer I ran into difficulties. The main one was that the EPROM pin (address 7) was defective and only one half of the memory was addressable. At first I assumed that my circuit was in error. After several days I convinced myself that the EPROM was bad and returned it for exchange and bought another chip as well. The exchanged chip had the same problem while the new chip appears to be working fine. (I also accidentally applied 25v to the 5v supply which fried 2 RAMS but in the process fortunately spared the rest of the computer.)

Trouble Shooting Before Putting the EPROM In

In SYNC (2:1) I showed a method of monitoring computer activities by injecting signals into the video output. This method is useful in dynamically testing the board and connection to the computer. With the computer on and the board connected, note the different patterns that the pins on the Z80 make relevant to the EPROM. For instance, the patterns for A₀ at the edge connector should be the same as A₀ at the EPROM.

Program 5. A Machine Language Program to be POKEd into the EPROM.

```
LD HL, (E-file)      ; Load HL with E-file location the end
                      ; of the variables list. A string must
                      ; be at the end of that list.

EX HL, DE
LD H, 0
LD L, 0
;Clear HL for addition

NEXT: DEC DE
LD A, (DE)
CMP, 129
JRC -NEXT
;See if it's a variable name

AGAIN: INC DE
LD A, (DE)
CMP, 01
RETZ
;If not check next byte

ADD HL, A
ADD HL, A
ADD HL, A
ADD HL, A
;Shift HL four places: 16 times.

SUB, 28
JRC, +ERROR
;Convert letter to number
;Error if string token less than 8
;See if token is greater than "f"
CMP, 16
JRC, +NO ERROR
;If not, jump to NO ERROR
RST 06, 9
;ERROR ends BASIC program with ERROR code "A"
LD C, A
NO ERROR: ADD HL, BC
JR AGAIN
;Add converted token to HL
;Do again
```

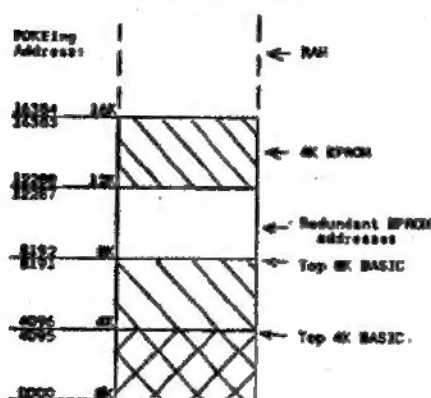
Decimal Noting for Program 6.

0-42	1-10	2-64	3-235	4-96
5-104	6-27	7-26	8-254	9-129
10-56	11-250	12-19	13-26	14-254
15-01	16-200	17-41	18-41	19-41
20-41	21-214	22-28	23-56	24-4
25-254	26-16	27-56	28-2	29-207
30-9	31-79	32-9	33-24	34-233

Chart 1.

MODE:	J1	J2	J3	J4
Test	in	out	in	out
Program	in	out	in	in
Read	out	in	out	out

Chart 2. Memory Locations.



The microprocessor starts at zero and proceeds up in addresses, doing each instruction in turn until it eventually loops back to a lower address.

Repeat for A₀ to A₁₁ and D₀ to D₇ as well. Make sure that there are five volts where there should be on the board. If all looks good, then you are ready to check the timing circuit.

Program 1 will allow you to set VR1 for timing the one-shot. First, put the programmer into the test mode (see Chart 1). Then run the program. It should take as close to 20 seconds to run as you can measure. Adjust VR1 until the program takes 20 seconds. If you cannot adjust the programmer to 20 seconds, then R may be changed, lower value to make it faster, higher value to slow it down.

Applying the 25 Volts

The 8K user should read the next section before going further.

Since the 25v can be so damaging, triple check that the 25v line only runs to pin 21 of the EPROM and that the diode is in properly and works properly. Then turn the power off.

Put the EPROM in, turn the power back on, and RUN Program 2. If Program 2 ran badly, go back and check the wiring. If it ran okay, then RUN Program 3. (4K ROM apply 25v before RUNNING the program; 8K ROM, during the run.) After the program is over and the 25v are removed, run Program 4. Program 4 checks to see that your wiring has not created redundant addressing in the memory. If the program prints more than one address, then most likely an address line is open (or the

EPROM has a defective address pin), and you must go back to the previous section and dynamically check the address and data lines. If program ran okay, then you are ready to program the EPROM with real data.

Verify that when the 25v are connected to the Vpp that only 3 to 30 ma are drawn by that pin. Also, if some other 25v source is used, make sure it is well regulated to within +1v of 25v.

Programming the EPROM

Programming is simple. Put the Eprom programmer into the program mode (Chart 1) and then POKE data into the EPROM addresses. Because of the way the 8K ROM uses the Refresh cycle, I have had trouble on occasion while the video is on. Therefore, for the 8K ROM the video must be "off" while the EPROM programmer is in the program mode. Program 5 is used for the 8K ROM. Do not have the 25v on when the 8K ROM video is

Program 7. Entering Hexadecimal Data.

```
10 FOR L=0 TO 10
20 PRINT L:
30 INPUT L:
40 PRINT L:
50 POKE L+H,USR N
60 NEXT L
```

Note: M = first address to which data is to be POKEd. N = address of Program 6 in the EPROM. This program cannot be used to POKE into the EPROM.



on, or else bad data may be written into the EPROM. Line 3 of the program is a loop which waits for the "P" key to be pressed. While the program is waiting, you must set the EPROM programmer to the "program" mode. Then press "P" and the program will POKE the data into the EPROM. You will know that the programming is done when the screen starts to glitter with garbage as the program gets SAVED in line 1000. Now place the EPROM into the "read" mode. Then press BREAK.

Two sample programs are shown which work for the 4K Basic. One converts a hexadecimal string into a number (Program 6). The other program converts a number into a hexadecimal representation which can be PRINTed (Program 8). Programs 7 and 9 show their uses in Basic programs.

Putting It to Work

This construction project is really an open ended affair, just as computers are. Once operating, your limits are your imagination, patience, and money. Though not particularly difficult to build, like all projects it will take fortitude to keep on going when things look hopeless (and they will).

Published programs should be checked to see if any changes must be made when relocating them to the EPROM addresses. Relative jumps and calls to inside the program have to be changed. An example of one particular problem is illustrated by trying to adapt the

Program 8. A Machine Language Program to be POKEd into the EPROM.

```
LD HL,(VARS)      ;Load HL with location of first Variable.
                   ;Variable must be there from BASIC
                   ;program and must be a one letter name.
                   ;Get past variable name.

INC HL
LD A,(HL)
LD B,A
AND A,$Fh          ;Mask last four bits
ADD A,28           ;Convert to number token
LD HL,(E-file)     ;Load HL with the address of the end of
                   ;the variable list. A two character
                   ;string must be there from the BASIC
                   ;program.

DEC HL
DEC HL
DEC HL
LD (HL),A
DEC HL
LD B,A
AND A,$Fh
RR
RR
RR
RR
ADD A,28
LD (HL),A
LD HL,$0000
RET

;Move past quote in string to
;last character in string
;Load converted token into string
;Move pointer to first character in string.
;Get rest of number.
;Mask upper four bits.
;Shift upper bits into lower bits
;Convert to number token.
;Place in upper character of string.
;Clear HL just for neatness' sake.
;Return to BASIC with the last string
;being a hexadecimal representation
;of the number in the first variable.
;Note: Only numbers 0 to 255 are converted.
```

"Scroll" routine of Logan (SYNC 1:4). The routine requires three bytes of RAM, one to pass the "Leave" parameter, the other two for an internal program register. The addresses of the RAM bytes were immediately before the subroutine. Obviously, an EPROM can not be used for RAM. My solution was to pass the "Leave" parameter through the RANDOM-ISE command and use the Syntax pointer location for the INTERNAL PROGRAM register since no Syntax pointing is required during the subroutine.

There have been many good subroutines in SYNC and the general techniques outlined in such articles can be used as long as they are not used blindly but understood and modified as needed. A good book on Z80 machine language programming will help a great deal.

A log of the addresses should be kept so that you know where what is and that you do not over write onto existing programs. Also tape copies of all programs should be kept in case it becomes necessary to reprogram the whole EPROM.

I have run my programmer with the power supply provided with the computer. However, when I started to add other devices, it quickly became necessary to supplement the power supply. If your system is already taxing the power supply, you will have to consider expanding it.

Construction Outline

- 1) Gather parts into one place and fit them into the PC board. Use the IC socket for the EPROM.
- 2) Solder the circuit together.
- 3) Test the board for proper connections.
- 4) Check and adjust the timing resistor. Check the 25v line.
- 5) Place the EPROM in and test it with the various programs (2 through 4).
- 6) If it is okay, program in some simple routines and see if they work.

A Cheap and Simple EPROM Programmer (3:1)

The author asks to have added at the end of the section "How the EPROM Programmer Works," p. 80, the following paragraph:

As designed, the EPROM programmer will not work with 64K memory modules. The reason is that they generate their own ROM C/S signals which would conflict with those generated by the programmer. However, once programmed, the EPROM could be placed into a ROM socket, provided that the pin configuration is compatible or is made compatible. The problem here is that no one EPROM pin assignment scheme exists. To make a "standard" ROM socket compatible with the EPROM, the following modifications to the ROM circuit (not the programmer) are required: traces to pins 18 and 20 should be reversed. The trace to pin 21 should be cut and pin 21 should be tied to Vcc (+5V). (Note: I have reversed the usual assignments of pins 19 and 20 to aid in compatibility.) In all probability, the ROM circuit provided would require such modifications.

Program 9. Display of Data as Hexadecimal Numbers.

```
10 LET X=0
20 FOR L=0 TO 10
30 LET X=PEEK (L+N)
40 LET M=X/16
50 PRINT L;"-";X;
60 NEXT L
```

Note: Line 5 reserves the first variable in the variable list for the subroutine. This must be done before the FOR command sets up its variable. The FOR variable cannot be used for the first variable, nor can variables with more than one character in the name. Line 30 makes sure that there is a two character string at the end of the variables list. M = first address of the area to be displayed. N = address of Program 8 in the EPROM.

Decimal listing for Program 8.

0-42	1-8	2-64	3-35	4-126
5-71	6-230	7-15	8-198	9-28
10-42	11-10	12-64	13-43	14-43
15-43	16-119	17-43	18-120	19-230
20-240	21-15	22-15	23-15	24-15
25-198	26-28	27-119	28-33	29-0
30-0	31-201			

ZX81 Software



```

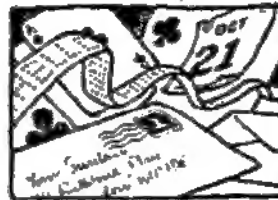
1 REM ZX-81/TS1000/TS1500
2 SLOW
3 REM NEVER PRESS RUN,BUT USE
GOTO 1 INSTEAD
5 REM U.3,PARTS.B3,PARTSDATA-
BASE,COPYRIGHT BY BILL HAMMER,
OTTAWA,1986,PLACED IN PUBLIC DOM
AIN 1999
35 LET T$="PARTS DATABASE"
55 REM GO TO STARTING DISPLAY
60 GOSUB 9600
65 REM GO TO INTRO INFO
70 GOSUB 9850
80 CLS
100 GOSUB 9600
110 PRINT "MENU"
115 PRINT
120 PRINT "1-START A NEW DATABA
SE"
125 PRINT
130 PRINT "2-ADD TO AN EXISTING
DATABASE"
135 PRINT
140 PRINT "3-SCAN DATA"
145 PRINT
150 PRINT "4-SAVE PROGRAM WITH
DATA"
155 PRINT
160 PRINT "5-QUIT PROGRAM"
165 PRINT
170 PRINT "6-LIST DATA"
185 PRINT
195 PRINT AT 20,0;"ENTER THE NU
MBER OF YOUR CHOICE"
200 INPUT MENU
210 CLS
220 IF MENU=1 THEN LET S$="*"
250 GOTO MENU*1000
300 GOTO 80
370 CLS
390 PRINT "THATS ALL FOLKS"
992 PRINT
995 PRINT "PROGRAM USED=";PEEK
16404+256;PEEK 16405-16384;" BYT
E$"
999 STOP
1025 LET P$=""
1030 GOSUB 9800
1015 PRINT AT 20,0;"ENTER ""QUIT
"" TO GO TO MENU"
1020 PRINT AT 4,0;"ENTER PARTS N
UMBER-MAX 12 DIGITS"
1025 PRINT AT 21,0;"
1025 IF P$<>"" THEN PRINT AT 17,
0;"PREVIOUS PART NO.";P$
1030 INPUT N$
1035 IF N$="QUIT" THEN GOTO 1400
1040 IF LEN N$>12 THEN GOSUB 110
0
1045 IF LEN N$>12 THEN GOTO 1030
1050 IF LEN N$<12 THEN GOSUB 120
0
1055 PRINT AT 20,0;"
1060 PRINT AT 4,0;"PARTS NUMBER=
";N$;
1065 PRINT AT 21,0;"
1070 PRINT AT 6,0;"ENTER QUANTIT
Y (UP TO 999)"
1080 INPUT Q$
1082 IF LEN Q$=2 THEN LET Q$="0"
+Q$
1083 IF LEN Q$=1 THEN LET Q$="00
"+Q$
1085 IF VAL Q$>999 OR VAL Q$<0 T
HEN PRINT AT 21,0;"UNACCEPTABLE
QUANTITY-RE-ENTER"
1090 IF VAL Q$>999 OR VAL Q$<0 T
HEN GOTO 1080
1091 PRINT AT 6,0;"QUANTITY=";Q$
1092 PRINT AT 20,0;"

```

```

1093 GOSUB 9800
1094 LET S$=S$+N$+Q$+"*"
1095 LET P$=N$
1096 GOTO 1010
1100 REM TOO LONG PART NUMBER
1130 PRINT AT 21,0;"PART NUMBER
TOO LONG-RE-ENTER"
1140 PRINT AT 4,0;"NO. ENTERED="
;N$;" IS TOO LONG"
1160 RETURN
1200 REM SUB TO LENGTHEN PARTS
NO. FIELD
1205 LET M$=""
1210 LET L=LEN N$
1220 FOR X=1 TO 12-L
1230 LET M$=M$+" "
1240 NEXT X
1250 LET N$=M$+N$
1260 RETURN
1400 GOTO 255
2000 REM ADD TO EXISTING DATABAS
E
2010 GOTO 1000
3000 REM DATA SCAN ROUTINE
3005 PRINT AT 17,0;"
3010 GOSUB 9600
3015 IF S$="*" THEN PRINT "NO DA
TA IN FILE"
3020 LET E=((LEN S$-1)/16)
3030 PRINT AT 19,0;"REPEAT,
TO GO TO MENU,""A"" TO REPEAT,"
"S"" TO STOP,""C"" TO CONT- INU
E,""D"" TO DELETE FILE"
3035 LET C=1
3040 FOR X=1 TO E
3042 PRINT AT 17,0;"
3045 PRINT AT 4,0;"FILE NO. ";C;
3050 LET K=((X-1)*16)+1
3055 IF S$(K+16)="*" THEN GOTO 3
150
3060 LET C=C+1
3060 PRINT AT 6,0;"PART NUMBER="
;S$(K+1 TO K+12)
3070 PRINT AT 8,0;"QUANTITY=";S$
(K+13 TO K+15)
3080 FOR Z=1 TO 35
3090 IF INKEY$="S" THEN GOSUB 33
00
3100 IF INKEY$="0" THEN GOTO 320
0
3110 NEXT Z
3150 NEXT X
3155 PRINT AT 17,0;"FILE END REA
CHED"
3160 IF INKEY$="0" THEN GOTO 320
0
3170 IF INKEY$="A" THEN GOTO 300
0
3180 GOTO 3160
3200 GOTO 255
3300 REM CONTROL SUBROUTINE
3310 IF INKEY$="C" THEN GOTO 333
0
3312 IF INKEY$="0" THEN GOTO 320
0
3313 IF INKEY$="A" THEN GOTO 300
0
3314 IF INKEY$="D" THEN LET S$(K
+16)="*"
3317 IF S$(K+16)="*" THEN PRINT
AT 17,0;"FILE DELETED"
3320 GOTO 3310
3330 RETURN
4000 REM SAVE PROGRAM WITH DATA
4040 CLS
4050 PRINT AT 9,4;"PRESS F10 FOR
RECORDING"
4060 PRINT AT 15,2;"START RESEET
RECORDED"
4070 PAUSE VAL "200"

```



```

4080 CLS
4090 SAVE "PARTS DB"
4095 GOTO 1
5000 REM QUIT PROGRAM
5010 GOTO 970
6000 REM LIST ROUTINE
6005 LET L=15
6010 IF S$="" THEN PRINT "NO DA
TA IN FILE"
6020 IF S$="*" THEN STOP
6025 REM CALCULATE NUMBER OF
PARTS NOS.
6030 LET E=((LEN S$-1)/16)
6035 IF E<16 THEN LET N=1
6036 IF E<16 THEN LET L=INT E
6040 IF E<16 THEN GOTO 6055
6050 FOR N=1 TO (INT (E/16)+1)
6055 CLS
6056 GOSUB 9800
6057 PRINT "PART NUMBER", "QUANTI
TY"
6060 FOR Z=1 TO L
6070 LET X=((N-1)*15)+Z
6080 LET K=((X-1)/16)+1
6085 IF K+16>LEN S$ THEN GOTO 61
35
6090 IF S$(K+16)="*" THEN GOTO 6
135
6100 PRINT S$(K+1 TO K+12), S$(K+
13 TO K+15)
6135 IF Z=L AND N=(INT (E/16)+1)
THEN GOTO 6170
6140 NEXT Z
6145 GOSUB 9500
6150 IF INT E<16 THEN GOTO 6170
6160 NEXT N
6170 PRINT "      NO MORE ITEMS"
6180 GOSUB 9500
6190 GOTO 255
9000 REM PROGRAM SAVE ROUTINE
9010 CLS
9015 CLEAR
9020 PRINT AT 9,4; "CASSETTE SAVE
PROGRAM"
9025 PRINT AT 15,2; "START CASSET
TE RECORDER NOW"
9030 PAUSE VAL "200"
9035 CLS
9040 SAVE "PARTS DB"
9050 GOTO 20
9099 STOP
9500 REM FLASH TO GO ON SUBR.
9505 PRINT AT 21,22; "
9510 PRINT AT 21,0; "PRESS ANY KE
Y TO
GO ON"
9520 FOR X=1 TO 3
9530 IF INKEY$(X) THEN GOTO 958
0
9535 NEXT X
9540 PRINT AT 21,0; "PRESS"
9550 FOR X=1 TO 3
9555 IF INKEY$(X) THEN GOTO 958
0

```

```

9560 NEXT X
9570 GOTO 9510
9580 CLS
9590 RETURN
9600 REM INITIAL DISPLAY
9630 PRINT AT 7,0; "
9640 PRINT AT 8,0; "
9650 IF LEN T$>28 THEN PRINT "CH
ANGE TITLE PRINT LINES 9650-9700
"
9655 PRINT "I"
9660 PRINT AT 9,INT ((30-LEN T$)
/2); T$
9670 PRINT AT 9,31; "I"
9720 PRINT AT 10,0; "
9730 PRINT AT 11,0; "
9740 PRINT AT 15,5; "PROGRAM (REV
,1990) BY      BILL HARNER,
OTTAWA"
9760 PRINT AT 21,0; "PRESS ""ENTE
R"" TO GO ON"
9765 INPUT A$
9770 CLS
9780 RETURN
9800 REM TITLE PRINT SUBROUTINE
9810 PRINT AT 0,INT ((32-LEN T$)
/2); T$
9815 PRINT
9820 RETURN
9850 REM PRELIMINARY INFO HERE
9860 GOSUB 9800
9870 PRINT "PARTS DATABASE IS A
SIMPLE DATA-BASE WHICH WILL HOLD
A PARTS      NUMBER UP TO 12 DIGI
TS LONG AND A 3 DIGIT QUANTITY F
OR EACH PART"
9875 PRINT
9880 PRINT "THIS DATABASE MAY PR
OVE USEFUL      FOR PARTS INVENTORY
OR AS A      BUILDING BLOCK TO AD
D TO TO MAKE MORE SOPHISTICATED
DATABASE"
9885 PRINT
9890 PRINT "THIS DATABASE VERSIO
N DOES NOT      HAVE A SORT OR SEARC
H FACILITY      BUT DOES HAVE A RECO
RD DELETE      ABILITY"
9960 GOSUB 9500
9990 RETURN
9991 REM NOTES
9997 REM SUBROUTINES: 9600 START-
ING DISPLAY      9500 FLASH TO GO
ON      9800 PAGE TOP TITLE
9955 TITLE REPEAT PAGE

```





LARKEN SYSTEM

DISK DRIVE MENU LOADER



EASY TO USE AND CONVEIENT

Just place the highlighted bar on the program name you wish to LOAD. This program uses the CAT feature to assemble the list of the programs to load. It can handle 100 titles, and you can go back and forth over the entire list of names.

The program was written for L. S. T. (Long Island Sinclair Timex) group by Keith Skapinski and is available on a 5 1/4 in. disk if you prefer. Order from Tom Skapinski, 7 Atkinson Lane, Coram, NY 11727 (Cost \$5.00).

INSTRUCTIONS:
Enter the first listing then compile using a Timachine by Novelsoft. Save as directed by Timachine. Name "DISKME.C1" then reloaded at 50000, and save as "DISKME.C1" CODE 50000,3745. Enter the second listing, the basic loader. Run; then break and goto 200 to save the two parts. I hope you enjoy using this program. Keep Computing!

```

100 REM USR 50000
101 REM OPEN #
102 CLS
103 FOR I=VAL "63000" TO VAL "63077"
104 READ A: POKE I,A: NEXT I
105 DATA VAL "195",VAL "43",VAL "246",VAL "196",VAL "72",VAL "245",VAL "195",VAL "104",VAL "245",VAL "243",VAL "205",VAL "0",VAL "201",VAL "206",VAL "100",VAL "251",VAL "201",VAL "206",VAL "33",VAL "246",VAL "58",VAL "176",VAL "32",VAL "50",VAL "29",VAL "32",VAL "205",VAL "123",VAL "33",VAL "112",VAL "32",VAL "17",VAL "156",VAL "224",VAL "176",VAL "20",VAL "237",VAL "176",VAL "195",VAL "38",VAL "245",VAL "205",VAL "33",VAL "120",VAL "246",VAL "38",VAL "176",VAL "32",VAL "50",VAL "29",VAL "32",VAL "205",VAL "156",VAL "130",VAL "17",VAL "112",VAL "32",VAL "1",VAL "20",VAL "237",VAL "176",VAL "205",VAL "150",VAL "140",VAL "0",VAL "206",VAL "126",VAL "205",VAL "120",VAL "195",VAL "26",VAL "246",VAL "200 REM PROGRAM
201 POKE VAL "23723",0: LET 0=0
202 BORDER 0: PAPER 0: INK 7: DIM A$(VAL "100",VAL "10"): DIM I(VAL "100")
220 CLS: PRINT INVERSE 1;"... I DISK MENU PROGRAM
NVERSE 0;"
230 RANDOMIZE USR VAL "63000"
235 LET name=VAL "57633"
240 FOR I=PI/PI TO VAL "100"
250 IF PEEK (name+VAL "3")=0 THEN GO TO 300
255 IF PEEK (name+VAL "3")=VAL "254" THEN LET name=name+34: GO TO 250
270 FOR J=PI/PI TO VAL "9": LET A$(I,J)=CHR$(PEEK (name+J)): NEXT J
275 LET I(I)=PEEK (name+11)
2800 LET name=name+34
2900 NEXT I
3000 FOR I=VAL "1" TO VAL "100": IF A$(I,VAL "1")="" THEN GO TO VAL "320"
310 NEXT I

```

This prog. utilizes CODE written by ?



```

320 LET P=1-VAL "1"
510 PRINT "PI/PI: AT 0,0: INVERS
PI/PI: (SPACE)=MOVE CURSOR (EM
TER)=LOAD (7)=PG UP (8)=BREAK (
6)=PG DOWN"
520 LET X=P/VAL "20": LET X=VAL
"41" AND X<0 AND X<=1)+(2 AND X)
1 AND X<=2)+(3 AND X>2 AND X<=3)
)+(4 AND X>3 AND X<=4)+(5 AND X>4)
530 LET C=PI/PI
535 LET d=2
540 PRINT AT 2,0: IF X>C THEN
FOR I=VAL "(C*20)-19" TO VAL "C*
20"
550 IF X=C THEN FOR Y=VAL "(C*2
0)-19" TO P
560 PRINT TAB VAL "11";A$(I),
570 NEXT I
580 IF X=C THEN FOR I=1 TO ((C*
20)-P): PRINT "... NEXT I
590 PRINT AT d,11: OVER 1: INVE
RSE 1:
600 PAUSE 0: LET q$=INKEY$: IF
q$="" THEN GO TO 600
602 IF q$="B" OR q$="b" THEN ST
OP
605 IF q$="" THEN PRINT AT d,1
1: OVER 1: INVERSE 1:
610 IF q$="" THEN LET d=d+1
620 IF d=22 THEN LET d=2
625 IF C=X AND d=22-((C*20)-P)
THEN LET d=2
630 IF q$="" THEN GO TO 590
635 IF q$="6" THEN LET C=C+1
640 IF C>X THEN LET C=X: BEEP .
01,10: GO TO 600
650 IF q$="7" THEN LET C=C-1
660 IF C=0 THEN LET C=1: BEEP .
01,10: GO TO 600
665 IF q$="6" OR q$="7" THEN LE
T C=VAL "2"
670 IF q$=CHR$(13) THEN GO TO 69
0
680 GO TO 540
690 LET I$=A$(((C*20)-19)+d-2)
695 FOR I=1 TO 10: IF I$(I)=""
THEN LET I$=I$(TO I-1): GO TO
700
697 NEXT I
700 CLS: PRINT "LOADING ";I$
705 FOR I=55368 TO 65367+LEN I$
POKE I,CODE L$(I-55367): NEXT
I: POKE I,CODE L$
740 REM CLOSE #
9000 CLEAR: PRINT #4: SAVE "DTS
KME.B1" LINE 1
10 REM DISK MENU LOADER
15 RANDOMIZE USR 100: OPEN #4,
"d:
20 CLS
30 PRINT #4: LOAD "DISKME.C1"O
ODM 50000
40 RANDOMIZE USR 50000
50 IF PEEK 55368=0 THEN GO TO
1200
60 LET I$="": LET L=55368
70 IF PEEK L=42 THEN GO TO 90
80 LET L=L+CHR$(PEEK L): LET
L=L+1: GO TO 70
90 IF L$(LEN I$-1)="C" THEN PR
INT #4: LOAD I$CODE: STOP
100 PRINT #4: LOAD I$
110 STOP
120 CLS
130 PRINT "B=Break program and
go to BASIC C=Change current dri
ve number"
140 PAUSE 0: LET q$=INKEY$: IF
q$="" THEN GO TO 140
150 IF q$="B" OR q$="b" THEN ST
OP
160 IF q$<>"C" AND q$<>"C" THEN
GO TO 140
170 CLS: PRINT "Which drive (0
-3)?"
180 INPUT d
190 PRINT #4: GO TO d: GO TO 40
2000 REM SAVE
210 PRINT #4: SAVE "DISKME.B1"
LINE 10
220 PRINT #4: SAVE "DISKME.C1"O
ODE 50000,3745

```